

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 3月31日  
Date of Application:

出願番号 特願2003-096015  
Application Number:  
[ST. 10/C] : [JP 2003-096015]

出願人 株式会社エヌ・ティ・ティ・ドコモ  
Applicant(s):

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2004年 3月26日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫

出証番号 出証特2004-3025490

【書類名】 特許願  
【整理番号】 DCMH140833  
【提出日】 平成15年 3月31日  
【あて先】 特許庁長官 殿  
【国際特許分類】 H04L 9/00  
【発明の名称】 端末装置、端末装置の制御方法、プログラム及び通信方法  
【請求項の数】 13  
【発明者】  
【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内  
【氏名】 市川 裕一  
【発明者】  
【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内  
【氏名】 成瀬 直樹  
【発明者】  
【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内  
【氏名】 大井 達郎  
【発明者】  
【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内  
【氏名】 渡邊 信之  
【発明者】  
【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内  
【氏名】 服部 易憲

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 竹下 理人

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 西田 真和

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 浅井 真生

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 津田 雅之

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 富岡 淳樹

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 山田 和宏

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 神谷 大

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 鶯尾 諭

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 山根 直樹

**【発明者】**

【住所又は居所】 東京都千代田区永田町二丁目11番1号 株式会社エヌ  
・ティ・ティ・ドコモ内

【氏名】 村上 圭一

**【特許出願人】**

【識別番号】 392026693

【氏名又は名称】 株式会社エヌ・ティ・ティ・ドコモ

**【代理人】**

【識別番号】 100098084

【弁理士】

【氏名又は名称】 川▲崎▼ 研二

**【選任した代理人】**

【識別番号】 100111763

【弁理士】

【氏名又は名称】 松本 隆

**【手数料の表示】**

【予納台帳番号】 038265

【納付金額】 21,000円

**【提出物件の目録】**

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 端末装置、端末装置の制御方法、プログラム及び通信方法

【特許請求の範囲】

【請求項 1】 ソフトウェアを起動し実行する実行手段と、

前記ソフトウェアを実行することにより実現されるアプリケーションに与えられた権限の内容を示すパーミッション情報を記憶する記憶手段と、

前記ソフトウェアが起動される場合には前記パーミッション情報が有効であるか否かを外部装置に対して確認するチェック手段と、

前記チェック手段による確認の結果、前記パーミッション情報が有効であることが確認された場合には前記実行手段によるソフトウェアの起動を許可する一方、前記パーミッション情報が有効でないことが確認された場合には前記実行手段によるソフトウェアの起動を許可しない起動制御手段と

を備えた端末装置。

【請求項 2】 前記チェック手段による確認を行うべきか否かを判断する判断手段を備え、

前記判断手段によって前記確認を行うべきと判断された場合には、前記チェック手段が前記確認を行う請求項 1 記載の端末装置。

【請求項 3】 前記判断手段は、

前記ソフトウェアの起動回数をカウントする起動回数カウント手段と、

前記ソフトウェアの連続した起動回数に対してどのくらいの頻度で前記確認を行うべきかということを示す頻度情報を記憶する頻度情報記憶手段とを含み、

前記起動回数カウント手段によってカウントされた起動回数と、前記頻度情報記憶手段によって記憶された頻度情報に基づいて、前記チェック手段による確認を行うべきか否かを判断する請求項 2 記載の端末装置。

【請求項 4】 前記判断手段は、

前記ソフトウェアの起動間隔を計時する計時手段と、

どのくらいの時間間隔で前記確認を行うべきかということを示す間隔情報を記憶する間隔情報記憶手段と

を含み、

前記計時手段によって計時される起動間隔と、前記間隔情報記憶手段によって記憶された間隔情報に基づいて、前記チェック手段による確認を行うべきか否かを判断する請求項 2 記載の端末装置。

【請求項 5】 前記チェック手段によって前記確認ができなかった場合であっても前記起動制御手段が前記ソフトウェアの起動を許可する回数を示す猶予回数情報を記憶した猶予回数記憶手段を備え、

前記起動制御手段は、前記チェック手段によって前記確認ができなかった場合、前記猶予回数記憶手段によって記憶された猶予回数情報が示す回数分は前記ソフトウェアの起動を許可する請求項 1 記載の端末装置。

【請求項 6】 前記外部装置との間で無線を利用して通信を行う移動機である請求項 5 記載の端末装置。

【請求項 7】 新たな前記パーミッション情報を前記外部装置から取得し、取得したパーミッション情報に基づいて前記記憶手段に記憶されているパーミッション情報を更新する更新手段を備えた請求項 1 記載の端末装置。

【請求項 8】 前記更新手段は、

前記チェック手段が前記外部装置に対して前記確認を行う際に、前記記憶手段によって記憶されている前記パーミッション情報が更新された時を示す更新時情報を前記外部装置に通知し、

これに応じて、前記外部装置から送信されてくる新たなパーミッション情報を受信し、

受信したパーミッション情報に基づいて前記記憶手段に記憶されているパーミッション情報を更新する請求項 7 記載の端末装置。

【請求項 9】 前記実行手段が実行している前記ソフトウェアによって実現されるアプリケーションが前記パーミッション情報の内容に反する場合、前記実行手段による当該ソフトウェアの実行を終了させる終了手段を備えた請求項 1 記載の端末装置。

【請求項 10】 前記パーミッション情報は、前記端末装置内部或いは外部のハードウェア資源或いはソフトウェア資源の利用に関する情報、又は、ネット

ワーク資源の利用に関する情報である請求項1に記載の端末装置。

【請求項11】 端末装置が、ソフトウェアを実行することにより実現されるアプリケーションに与えられた権限の内容を示すパーミッション情報を外部装置から取得するステップと、

端末装置が、前記ソフトウェアを起動する場合に前記パーミッション情報が有効であるか否かを外部装置に対して確認するステップと、

端末装置が、前記パーミッション情報が有効であることを確認した場合には前記ソフトウェアの起動を許可する一方、前記パーミッション情報が有効でないことを確認した場合には前記ソフトウェアの起動を許可しないステップと

を備えた端末装置の制御方法。

【請求項12】 コンピュータに、

ソフトウェアを実行することにより実現されるアプリケーションに与えられた権限の内容を示すパーミッション情報を記憶手段に記憶させる機能と、

前記ソフトウェアが起動される場合に前記パーミッション情報が有効であるか否かを外部装置に対して確認する機能と、

前記確認の結果、前記パーミッション情報が有効であることが確認された場合には前記ソフトウェアの起動を許可する一方、前記パーミッション情報が有効でないことが確認された場合には前記ソフトウェアの起動を許可しない機能とを実現させるためのプログラム。

【請求項13】 アプリケーションを実現するためのソフトウェアを内包した実体情報を格納した情報提供サーバ装置と、端末装置が前記ソフトウェアを実行することにより実現されるアプリケーションに与えられた権限を示すパーミッション情報を内包したセキュリティ記述情報を格納した管理サーバ装置と、前記実体情報に依存した内容を有し前記実体情報の格納位置と前記セキュリティ記述情報の格納位置とが記述されたアプリケーション記述情報を格納した情報提供サーバ装置とを有した通信システムから、前記端末装置に対して、前記アプリケーション記述情報を送信するステップと、

前記端末装置が、前記通信システムから送信されてくるアプリケーション記述情報に内包されている前記セキュリティ記述情報の格納位置を前記通信システム

に通知するステップと、

前記通信システムが、前記通知されたセキュリティ記述情報の格納位置に基づいて、当該セキュリティ記述情報を前記端末装置に送信するステップと、

前記端末装置が、受信した前記セキュリティ記述情報を記憶するステップと、

前記端末装置が、受信した前記セキュリティ記述情報に内包されている前記実体情報の格納位置を前記通信システムに通知するステップと、

前記通信システムが、前記通知された実体情報の格納位置に基づいて、当該実体情報を前記端末装置に送信するステップと、

前記端末装置が、前記通信システムから送信されてくる実体情報に内包されるソフトウェアをインストールするステップと、

前記端末装置が、インストールされている前記ソフトウェアを起動する場合に、前記記憶しているセキュリティ記述情報が有効であるか否かを前記通信システムに対して確認するステップと

前記端末装置が、前記確認の結果、前記セキュリティ記述情報が有効であることが確認された場合には前記ソフトウェアの起動を許可する一方、前記パーミッション情報が有効でないことが確認された場合には前記ソフトウェアの起動を許可しないステップと

を備えた通信方法。

#### 【発明の詳細な説明】

##### 【0001】

###### 【発明の属する技術分野】

本発明は、アプリケーションに対し、その挙動に関する権限を与える技術に関する。

##### 【0002】

###### 【従来の技術】

移動機の高機能化が急速に進みつつある。最近では、Java（登録商標）プログラミング言語に従って記述されたプログラムを含むソフトウェア（以下、Java-APソフトウェアという）をネットワークを介してダウンロードし、これを起動・実行してJavaアプリケーション（以下、Java-APという）

を実現する機能を備えた移動機が開発されている。

### 【0003】

#### 【発明が解決しようとする課題】

ところで、移動機内に実現されるJava-APの挙動についての制限は、通信アプリケーションなどの移動機が元から備えているネイティブアプリケーションの挙動についての制限よりも厳しくなっている（例えば特許文献1参照）。例えばJava-APは、移動機内の電話帳データ等の秘匿性の高い情報を参照することができないようになっている。このような制限の相違により、悪意をもつて作成されたJava-AP或いは不具合を有するJava-APによって移動機内の秘匿性の高い情報が漏洩したり改竄されてしまうという事態を、確実に回避することができるようになっている。

### 【0004】

#### 【特許文献1】

特開平10-254783号公報

### 【0005】

しかし、上述した厳しい挙動制限を全てのJava-APに対して一律に課すだけでは、Java-APの高機能化や多様化を望むユーザやCP（Contents Provider）の意向に添うことができない。例えば、ある程度の信頼性が保証されるのであれば、移動機に格納された個人情報を参照する権限をJava-APに与えてもよいと考えるユーザは少なくないと想定される。一方、CP側にも、移動機に格納されている個人情報や、移動機が有する多数の機能の使用を前提とした、より魅力的なJava-APを提供したいという要望がある。

### 【0006】

これらの要望を満たす仕組みとして次のようなものが考えられる。

例えば移動機のユーザに対して通信サービスを提供する通信事業者が上述した挙動制限を緩和した権限をJava-APに与え、この権限を移動機に通知する。この通信事業者は、ユーザにとって信頼できる機関であるから、以下、信頼機関という。移動機はこの権限に基づいてJava-APの挙動を制限すればよい。

### 【0007】

しかしながら、ここで次のような問題が危惧される。

上述したように、信頼機関が挙動制限を緩和した権限を移動機に通知し、以降、移動機はこの権限に基づいて Java-AP の挙動を制限するわけだが、信頼機関が上記の通知後に権限を変更（権限自体の消滅や権限内容の変更を含む）したい場合がある。例えば、いったんは挙動制限を緩和した権限が与えられた Java-AP が、実は、ユーザの承諾なしに個人情報を移動機のメモリから読み出して他の外部装置に送信するというような、ユーザにとって不利益な動作を行うアプリケーションであることが発覚した場合等である。このような場合、信頼機関としては、早急にこの Java-AP の利用を差し止めたり、Java-AP に与えられた権限の内容を変更しなければならない。

### 【0008】

本発明は、上述した事情に鑑みて為されたものであり、アプリケーションに与えられた権限が変更された場合に、その変更後の権限内容を、移動機等の端末装置におけるアプリケーションに反映させるための仕組みを提供することを目的としている。

### 【0009】

#### 【課題を解決するための手段】

上述した課題を解決するために、本発明は、ソフトウェアを起動し実行する実行手段と、前記ソフトウェアを実行することにより実現されるアプリケーションに与えられた権限の内容を示すパーミッション情報を記憶する記憶手段と、前記ソフトウェアが起動される場合には前記パーミッション情報が有効であるか否かを外部装置に対して確認するチェック手段と、前記チェック手段による確認の結果、前記パーミッション情報が有効であることが確認された場合には前記実行手段によるソフトウェアの起動を許可する一方、前記パーミッション情報が有効でないことが確認された場合には前記実行手段によるソフトウェアの起動を許可しない起動制御手段とを備えた端末装置を提供する。

この端末装置によれば、ソフトウェアを起動する場合にパーミッション情報が有効か否かを外部装置に確認するので、起動すべきでないソフトウェアの起動を

制限することができる。

#### 【0010】

前記端末装置は、前記チェック手段による確認を行うべきか否かを判断する判断手段を備え、当該判断手段によって前記確認を行うべきと判断された場合には、前記チェック手段が前記確認を行うようにしてもよい。

この端末装置によれば、前記確認を行うべきか否かを判断するので、常に外部装置に対して確認を行う場合よりも、通信処理が簡素化される。

#### 【0011】

この場合、前記判断手段は、前記ソフトウェアの起動回数をカウントする起動回数カウント手段と、前記ソフトウェアの連続した起動回数に対してどのくらいの頻度で前記確認を行うべきかということを示す頻度情報を記憶する頻度情報記憶手段とを含み、前記起動回数カウント手段によってカウントされた起動回数と、前記頻度情報記憶手段によって記憶された頻度情報に基づいて、前記チェック手段による確認を行うべきか否かを判断するようにしてもよい。

これによって、ソフトウェアの起動回数に応じた確認を行うことができる。

#### 【0012】

また、前記判断手段は、前記ソフトウェアの起動間隔を計時する計時手段と、どのくらいの時間間隔で前記確認を行うべきかということを示す間隔情報を記憶する間隔情報記憶手段とを含み、前記計時手段によって計時される起動間隔と、前記間隔情報記憶手段によって記憶された間隔情報に基づいて、前記チェック手段による確認を行うべきか否かを判断するようにしてもよい。

これによって、ソフトウェアの起動間隔に応じた確認を行うことができる。

#### 【0013】

また、前記チェック手段によって前記確認ができなかった場合であっても前記起動制御手段が前記ソフトウェアの起動を許可する回数を示す猶予回数情報を記憶した猶予回数記憶手段を備え、前記起動制御手段は、前記チェック手段によって前記確認ができなかった場合、前記猶予回数記憶手段によって記憶された猶予回数情報が示す回数分は前記ソフトウェアの起動を許可するようにしてもよい。

特に端末装置が前記外部装置と間で無線を利用して通信を行う移動機であると

、無線通信に特有の圈外状態や通信障害によって前記確認ができない場合があるが、このような場合であっても猶予回数分はソフトウェアが起動されるので、ユーザにとって使い勝手がよい。

#### 【0014】

前記端末装置は、前記外部装置と間で無線を利用して通信を行う移動機であれば望ましい。

また、前記端末装置は、新たな前記パーミッション情報を前記外部装置から取得し、取得したパーミッション情報に基づいて前記記憶手段に記憶されているパーミッション情報を更新する更新手段を備えていてもよい。この場合、前記更新手段は、前記チェック手段が前記外部装置に対して前記確認を行う際に、前記記憶手段によって記憶されている前記パーミッション情報が更新された時を示す更新時情報を前記外部装置に通知し、これに応じて、前記外部装置から送信されてくる新たなパーミッション情報を受信し、受信したパーミッション情報に基づいて前記記憶手段に記憶されているパーミッション情報を更新するようにしてもよい。

また、前記端末装置は、前記実行手段が実行している前記ソフトウェアによって実現されるアプリケーションが前記パーミッション情報の内容に反する場合、前記実行手段による当該ソフトウェアの実行を終了させる終了手段を備えていてもよい。

また、前記パーミッション情報は、前記端末装置内部或いは外部のハードウェア資源或いはソフトウェア資源の利用に関する情報、又は、ネットワーク資源の利用に関する情報であってもよい。

#### 【0015】

また、本発明は、コンピュータに、ソフトウェアを実行することにより実現されるアプリケーションに与えられた権限の内容を示すパーミッション情報を記憶手段に記憶させる機能と、前記ソフトウェアが起動される場合に前記パーミッション情報が有効であるか否かを外部装置に対して確認する機能と、前記確認の結果、前記パーミッション情報が有効であることが確認された場合には前記ソフトウェアの起動を許可する一方、前記パーミッション情報が有効でないことが確認

された場合には前記ソフトウェアの起動を許可しない機能とを実現させるためのプログラムを提供する。このプログラムはコンピュータによって読みとり可能な記録媒体に記録されて提供され得る。

### 【0016】

また、本発明は、アプリケーションを実現するためのソフトウェアを内包した実体情報を格納した情報提供サーバ装置と、端末装置が前記ソフトウェアを実行することにより実現されるアプリケーションに与えられた権限を示すパーミッション情報を内包したセキュリティ記述情報を格納した管理サーバ装置と、前記実体情報に依存した内容を有し前記実体情報の格納位置と前記セキュリティ記述情報の格納位置とが記述されたアプリケーション記述情報を格納した情報提供サーバ装置とを有した通信システムから、前記端末装置に対して、前記アプリケーション記述情報を送信するステップと、前記端末装置が、前記通信システムから送信されてくるアプリケーション記述情報に内包されている前記セキュリティ記述情報の格納位置を前記通信システムに通知するステップと、前記通信システムが、前記通知されたセキュリティ記述情報の格納位置に基づいて、当該セキュリティ記述情報を前記端末装置に送信するステップと、前記端末装置が、受信した前記セキュリティ記述情報を記憶するステップと、前記端末装置が、受信した前記セキュリティ記述情報に内包されている前記実体情報の格納位置を前記通信システムに通知するステップと、前記通信システムが、前記通知された実体情報の格納位置に基づいて、当該実体情報を前記端末装置に送信するステップと、前記端末装置が、前記通信システムから送信されてくる実体情報に内包されるソフトウェアをインストールするステップと、前記端末装置が、インストールされている前記ソフトウェアを起動する場合に、前記記憶しているセキュリティ記述情報が有効であるか否かを前記通信システムに対して確認するステップと、前記端末装置が、前記確認の結果、前記セキュリティ記述情報が有効であることが確認された場合には前記ソフトウェアの起動を許可する一方、前記パーミッション情報が有効でないことが確認された場合には前記ソフトウェアの起動を許可しないステップとを備えた通信方法を提供する。

### 【0017】

### 【発明の実施の形態】

以下、図面を参照して、本発明の一実施形態について説明する。なお、図面において、共通する部分には同一の符号が付されている。

#### 【0018】

最初に、従来の移動機によるJava-APソフトウェアのダウンロード手順について説明すると、移動機は、まず、WWW (World Wide Web) を構成するサーバ装置からADF (Application Descriptor File) を取得し、次いでJar (Java Archive) ファイルを取得するという流れになる。

#### 【0019】

より具体的には、ADFはJarファイルに依存した内容となっており、例えば、Jarファイルの格納位置を示すURL (以下、パッケージURLという)、Jarファイルのサイズを示す情報、Jarファイルの最終変更日時を示す情報等を必須情報として内包している。ADFを取得した移動機は、このADFの内容を参照するとともに自機の空きメモリ容量を確認する等して、ダウンロードしようとしているJava-APソフトウェアを自機にインストール可能であるか否かを判断する。インストール可能と判断すると、移動機は、ADFに内包されているパッケージURLを用いて、WWWを構成するサーバ装置からJarファイルを取得する。このJarファイルには、Java-APソフトウェアが格納されており、Jarファイルの取得をもってJava-APソフトウェアのダウンロードは完了する。以後、移動機において、ダウンロードされたJava-APソフトウェアが起動可能に設定され、このJava-APソフトウェアのインストールが完了する。なお、これらのADF及びJarファイルはJava-APソフトウェアを提供するCPによって作成されるのが普通である。

#### 【0020】

一方、本実施形態におけるJava-APソフトウェアのダウンロードは、まず、移動機のユーザが所望するJava-APソフトウェアに対応したADFを移動機に配信し、次いで、上記Java-APソフトウェアに対応したSDF (Security Descriptor File;セキュリティ記述ファイル) と称せられるファイルを移動機に配信し、最後にJarファイルを移動機に配信するという手順で行わ

れる。即ち、ADF、SDF、Jarファイルという順で配信される。これらの各種ファイルのうち、ADF及びJarファイルは、従来と同じく、Java-APソフトウェアを提供するCPによって作成される。これに対し、SDFは、上記通信事業者とCPとの間でJava-APソフトウェアに関して締結された契約に従い、上記通信事業者によって作成されるようになっている。

### 【0021】

ここで、SDFとは、移動機内におけるJava-APに与えられた権限の内容が記述されたファイルであり、移動機はこのSDFの記述内容に従ってJava-APソフトウェアを実行するようになっている。ただし、このSDFは有効である状態と失効されている状態があり、この有効或いは失効の別はネットワーク上のサーバによって記憶されている。移動機は、インストールしたJava-APソフトウェアを起動する際には、まず、上記サーバにアクセスしてSDFの有効或いは失効の状態の別を確認しなければならない。以下、この確認処理を「SDFチェック」という。このSDFチェックの結果、SDFが有効な状態であれば、移動機は、そのSDFの記述内容に従ってJava-APソフトウェアを起動し、実行する。一方、SDFが失効されている状態であれば、移動機は、Java-APソフトウェアを起動することができないようになっている。

### 【0022】

このようにSDFの有効或いは失効の別をネットワーク上で設定することができる、通信事業者は、不適切なJava-APソフトウェアに対応するSDFを失効状態に設定するだけで、その設定処理以降は、移動機におけるJava-APソフトウェアの起動を防止することができる。

ここでいう不適切なJava-APソフトウェアとは、例えば、ユーザの承諾なしに重要な情報を移動機のメモリから読み出して他の外部装置に送信する等の、ユーザにとって不利益な動作を移動機に実行させるソフトウェアである。通信事業者はこのようなJava-APソフトウェアの存在を知ったときには、そのJava-APソフトウェアに対応するSDFを失効状態に設定すればよい。また、CPと通信事業者との間で締結された契約が無効となったときにも（例えば契約期間が経過したとか、CPが契約に定められた料金を通信事業者に支払わな

かったとき）、通信事業者はこのようなC Pが提供するJava-A Pソフトウェアに対応するS D Fを失効状態に設定すればよい。また、例えば、Java-A Pソフトウェアが試用のソフトウェアであるような場合においても、その試用期間が過ぎれば、対応するS D Fを失効状態に設定すればよい。

### 【0023】

また、逆に、失効状態に設定していたS D Fを有効状態に設定することにより、そのS D Fに対応したJava-A Pソフトウェアの利用を再開するようにしてもよい。

### 【0024】

#### (1) 構成

次に、本実施形態に係る通信システムの構成について説明する。

図1に示されるように、この通信システムは、インターネット11に接続されたC Pサーバ装置13と、通信事業者が移動パケット通信サービスを提供するために用いる移動パケット通信網15と、この移動パケット通信網15との間で無線パケット通信を行ってこの移動パケット通信網15を介して通信相手とパケット通信を行う移動機16と、インターネット11と移動パケット通信網15とを相互接続するゲートウェイサーバ装置17と、専用線によりゲートウェイサーバ装置17に接続された管理サーバ装置18とを有する。なお、この通信システムには多数の移動機やC Pサーバ装置が存在するが、図面が繁雑になるのを避けるために一つの移動機16と1つのC Pサーバ装置13のみが図示されている。

### 【0025】

以下、この通信システムの各構成要素について詳細に説明する。

#### (1-1) C Pサーバ装置

C Pサーバ装置13は一般的なWWWサーバ装置と同様のハードウェアおよび機能を有している。C Pサーバ装置13の不揮発性メモリ13Aには、Javaプログラミング言語によって作成されたプログラムを含む複数のJ arファイルと、このJ arファイルに関する情報が記述された複数のA DFとが記憶されている。

### 【0026】

ここで、本実施形態では、対応するSDFが存在するJava-APソフトウェアと、対応するSDFが存在しないJava-APソフトウェアとが用意されている。前者のJava-APソフトウェアは、対応するSDFに記述された挙動制限を受けるものであり、通信事業者がCPとの契約に基づいて信頼性を保証したものであることから「トラステッドJava-APソフトウェア」と呼ぶ。後者は、要するに従来のJava-APソフトウェアであるが、本実施形態では「非トラステッドJava-APソフトウェア」と呼ぶ。CPサーバ装置13によって記憶されるADFには、トラステッドJava-APソフトウェアに対応したADFと、非トラステッドJava-APソフトウェアに対応したADFとがある。

### 【0027】

これらのいずれのADFにおいても、WWWにおけるJarファイルの記憶位置を示すパッケージURLや、Jarファイルのサイズを示す情報や、Jarファイルの最終変更日時を示す情報等の、周知のADFに含まれているものと同一の情報が記述されている。さらに、トラステッドJava-APソフトウェアに対応したADFには、上に列挙したパッケージURL等の他、トラステッドJava-APソフトウェアを一意に識別するための識別情報である「APID」と、SDFがWWWにおいて記憶されている位置を示すURL（以下、「SDF-URL」と呼ぶ）とが記述されている。移動機16はCPサーバ装置13からトラステッドJava-APソフトウェアに対応するADFを取得し、次いで、このADFに記述されているSDF-URLに基づいてSDFを取得することができる。

なお、本実施形態の説明において「Java-APソフトウェア」と言う場合には、それが「トラステッドJava-APソフトウェア」であればADF、SDF及びJarファイルを含む概念とし、「非トラステッドJava-APソフトウェア」であればADF及びJarファイルを含む概念とする。

### 【0028】

#### (1-2) ゲートウェイサーバ装置

ゲートウェイサーバ装置17は前述の通信事業者により管理されており、移動

パケット通信網 15 とインターネット 11 とを接続する一般的なゲートウェイサーバ装置と同様の構成を有し、移動パケット通信網 15 とインターネット 11 と管理サーバ装置 18 との間で相互に通信を中継する。

### 【0029】

#### (1-3) 管理サーバ装置

管理サーバ装置 18 は前述の通信事業者により管理されており、一般的な WWW サーバ装置と同様のハードウェアおよび機能を有する。管理サーバ装置 18 の不揮発性メモリ 18A には、複数のトラステッド Java-AP ソフトウェアにそれぞれ対応した複数の SDF が記憶されている。これらの SDF には、前述したように、トラステッド Java-AP ソフトウェアに与えられた権限の内容が記述されており、さらに、その SDF 自体が有効であるか或いは失効しているかという情報が記述されている。移動機 16 は、トラステッド Java-AP ソフトウェアを起動する際にはこの管理サーバ装置 18 にアクセスして SDF の有効或いは失効の別を確認することによって、その起動処理の可否を判断するようになっている。

### 【0030】

図 2 は、SDF に記述されている内容が、管理サーバ装置 18 から移動機 16 に通知されるときの HTTP メッセージの一例を示した図である。

図 2 に示すフィールド「Content-Type」は HTTP メッセージのエンティティボディのタイプを示すフィールドであるが、ここでは、エンティティボディが SDF に関するものであることを意味する「application/x-sdf」がフィールド値として記述されている。

### 【0031】

パラメータ「Sts」は、SDF の有効或いは失効の状態の別を示すものであり、以降、「SDFステータス」という。この SDF ステータスが「00」であるときは SDF が有効であることを示し、「10」であるときは SDF が失効していることを示している。図 2 の例では パラメータ「Sts」のパラメータ値が「00」であるので、この SDF は有効であることを意味している。また、パラメータ「PackageURL」は、前述した ADF に記述されるパッケージ URL と同じものであ

る。

### 【0032】

次に、パラメータ「CheckCnt」は、トラステッドJava-APソフトウェアの連続した起動回数に対してSDFチェックをどのくらいの頻度で行うべきかということを示すものであり、以降、「SDFチェック頻度」という。換言すれば、このSDFチェック頻度は、移動機16がSDFチェックを行わずに連続してトラステッドJava-APソフトウェアを起動することができる回数を示している。本実施形態では、このSDFチェック頻度として、1回から999回まで設定可能である。図2に示す例では、パラメータ「CheckCnt」のパラメータ値が「005」であるので、移動機16は連続して5回まではSDFチェックを要しないでトラステッドJava-APソフトウェアを起動可能であることを意味している。

### 【0033】

次に、パラメータ「CheckInt」はSDFチェックをどのくらいの時間間隔で行うべきかということを示すものであり、以降、「SDFチェック間隔」という。より具体的には、このSDFチェック間隔は、移動機16が前回SDFチェックを行った時点から次のSDFチェックを行わなくてもトラステッドJava-APソフトウェアを起動可能な日数を示しており、本実施形態では1日から999日まで設定可能である。図2に示す例では、パラメータ「CheckInt」のパラメータ値が「020」であるので、移動機16は、トラステッドJava-APソフトウェアを起動する際に一度SDFチェックを行うと、それ以降、20日を経過しない間に同じトラステッドJava-APソフトウェアを起動する場合にはSDFチェックを要しないが、20日を経過してから再度同じトラステッドJava-APソフトウェアを起動する場合にはSDFチェックを行う必要があることを意味している。

### 【0034】

次に、パラメータ「SuspendedCnt」は、移動機16がSDFチェックを行うことが不可能であった場合でも、トラステッドJava-APソフトウェアを起動することが可能な回数を示すものであり、以降、「SDFチェック猶予回数」と

いう。

移動機 16 は無線通信によって管理サーバ装置 18 にアクセスし、 SDF チェックを行うようになっている。しかしながら、例えば移動機 16 が移動パケット通信網 15 のサービスエリア外に所在しているとき（いわゆる圏外状態のとき）や、無線通信においては発生しがちな通信障害時には、 SDF チェックを行うことができなくなってしまう。これではユーザは好きなときにトラステッド Java - AP ソフトウェアを起動できない恐れが多分にあり、非常に不便である。そこで、本実施形態では、上記のような圏外状態や通信障害によって SDF チェックが不可能な場合であっても、ある程度の回数は SDF チェックを要せずにトラステッド Java - AP ソフトウェアの起動を認めるようにしている。本実施形態における SDF チェック猶予回数としては 1 回から 999 回まで設定可能である。図 2 に示した例では、パラメータ「SuspendedCnt」のパラメータ値が「005」であるので、移動機 16 は SDF チェックが不可能であっても連続 5 回まではトラステッド Java - AP ソフトウェアを起動することができることを意味している。

### 【0035】

次に、パラメータ「Lmd」は、管理サーバ装置 18 において SDF が最後に更新された日時を示したものであり、以下、「最終更新日時」という。図 2 に示す例では、パラメータ値が「20020614120552」となっているが、これは SDF の最終更新日時が 2002 年 6 月 14 日 12 時 5 分 52 秒であることを意味している。この最終更新日時は、管理サーバ装置 18 が更新された SDF の内容を移動機 16 に反映させるか否かを判断するために利用される。

### 【0036】

次に、パラメータ「GetPrivateInfo」、「UserMail」、「MessageApp」、「SetPhoneTheme」、「SetLaunchTime」、「AllowedHost」は、移動機 16 におけるトラステッド Java - AP ソフトウェアに与えられた権限の内容を示す情報であり、以下、パーミッション情報という。

例えば、パラメータ「GetPrivateInfo」は、移動機 16 に格納された電話帳データや未読電子メール等の個人的な情報を参照するときに必須とされるトラステ

ツドA P I (Application Programming Interface) の使用可否を示したものである。使用が可の場合には、パラメータ値「Yes」が記述される。同様に、パラメータ「UserMail」、「MessageApp」、「SetPhoneTheme」、「SetLaunchTime」については、それぞれ対応するトラステッドA P Iを使用することが許可されている場合に、パラメータ値「Yes」が設定される。図2に示す例では、いずれのパラメータについてもパラメータ値も「Yes」が設定されているので、トラステッドJ a v a - A Pは各種トラステッドA P Iを使用することができることになる。

### 【0037】

次に、パラメータ「AllowedHost」は、移動機16がトラステッドJ a v a - A Pソフトウェアを実行している間にアクセス可能な通信装置のU R Lを示したものであり、以下、アクセス許可U R Lリストという。

移動機16にダウンロードされたJ a v a - A Pソフトウェアは、一般に、サンドボックスと呼ばれるセキュリティモデルに従う。このサンドボックスによれば、移動機16がJ a v a - A Pソフトウェアを実行している間は、その移動機16の通信相手が上記J a v a - A Pソフトウェアのダウンロード元のサーバのみに厳格に制限される。このような厳しい制限の下では、ユーザに対して多様なアプリケーションを提供しづらくなってしまう。そこで、本実施形態では、トラステッドJ a v a - A Pソフトウェアに対しては、ダウンロード元サーバに加えて、予め定められた通信装置との間での通信を許可している。この通信が許可された通信装置のU R Lがパラメータ「AllowedHost」のパラメータ値として設定されている。図2に示す例では、「http://aaa.co.jp」と「http://bbb.co.jp」のいずれかを含むU R Lを有する通信装置との間でポート番号「8080」を利用した通信が許可されていることを意味している。

### 【0038】

なお、このアクセス許可U R Lリストのパラメータ値として「any」が設定されている場合には、トラステッドJ a v a - A Pソフトウェアを実行する移動機16はあらゆる通信装置と通信が可能であることを意味している。ただし、このように通信相手を全く自由にしてしまうとセキュリティが確保できない恐れもある。

るので、特別に通信を許可しない通信装置のURLを指定することもできる。このようなときには、パラメータ「DisallowedHost」（以下、このパラメータをアクセス禁止リストという）のパラメータ値に、通信が許可されていない通信装置のURLを記述すればよい。

### 【0039】

#### （1-4）移動機

移動機16は、図3に示されるように、OS（オペレーティングシステム）ソフトウェア、Java-APを実行する環境を構築するためのJava環境ソフトウェアおよび各種ネイティブAPソフトウェア等を記憶したROM16Aと、ROM16Aからプログラムを読み出して実行するCPU16Bと、表示部16Cと、不揮発性メモリ16Dと、RAM16Eと、通信部16Fと、操作部16Gと、計時部16Hとを有し、これらは通信線によって接続されている。

### 【0040】

表示部16Cは、例えば液晶表示パネルやパネル駆動回路を有し、CPU16Bから供給されるデータで表される画像を表示する。通信部16Fは、アンテナや無線送受信部を備え、移動パケット通信網15と無線パケット通信を行うものであり、CPU16Bと移動パケット通信網15との間でパケットを中継する。また、通信部16Fは、通話のためのCODECやマイク、スピーカ等をも備えており、これによって移動機16は図示せぬ移動電話網を介して回線交換による通話を行うこともできる。操作部16Gは操作子を備え、操作子の操作に応じた信号をCPU16Bへ供給する。計時部16Hは現在の年月日及び時刻（以下、単に現在日時という）を計時する。なお、計時部16Hがより正確な現在日時を計時するためには、例えば移動パケット通信網15の図示せぬ基地局から制御チャネルを用いて定期的に通知される現在日時に同期させるような処理を行ってもよい。

### 【0041】

不揮発性メモリ16Dは例えばSRAM（Static Random Access Memory）やEEPROM（Electrically Erasable and Programmable Read Only Memory）である。また、この不揮発性メモリ16Dは、WWWを構成するサーバ

装置からダウンロードしたJava-APソフトウェアを記憶するために使用される。

#### 【0042】

ここで、トラステッドJava-APソフトウェアのADF及びSDFに関して不揮発性メモリ16Dに記憶される内容について説明する。

図4は、トラステッドJava-APソフトウェアのADF及びSDF（図2に示したもの）の内容に基づいて、不揮発性メモリ16Dに記憶された内容を示している。さらに、この不揮発性メモリ16Dには、図4に示すように、SDFチェック頻度に対応して「起動回数」が、SDFチェック間隔に対応して「経過時間」が、SDFチェック猶予回数に対応して「猶予回数」が記憶されている。

起動回数とは、移動機16が前回SDFチェックを行ってから次のSDFチェックを行うことなくトラステッドJava-APソフトウェアを起動した回数である。この起動回数がSDFチェック頻度（図4では5回）に到達すると、その次にトラステッドJava-APソフトウェアが起動される際にはSDFチェックが必要と判断される。

#### 【0043】

次に、経過時間とは、移動機16が前回SDFチェックを行ってから経過した時間（日数）であり、前述した計時部16Hによって計時される。この経過時間がSDFチェック間隔（図4では20日）に到達すると、その次にトラステッドJava-APソフトウェアが起動される際にはSDFチェックが必要と判断される。そして、猶予回数とは、移動機16がSDFチェックが不可能であったにもかかわらずJava-APソフトウェアが起動された回数であり、この猶予回数がSDFチェック猶予回数（図4では5回）に到達すると、それ以後は、SDFチェックが行われない限りトラステッドJava-APソフトウェアは起動されない。

#### 【0044】

さて、移動機16の図示せぬ電源が投入されると、CPU16BはRAM16Eをワークエリアとし、ROM16AからOSソフトウェアを読み出してこれを実行する。これによって、CPU16Bは移動機16内にて図5に示すOSを実

現する。CPU16BはOSを実現することによって、操作部16Gから供給される信号に基づきユーザの指示を特定し、この指示に応じた処理を行う。

#### 【0045】

なお、以下では説明の便宜のため、CPU16BがOSソフトウェアを実行することによって行う動作は、「OS」をその動作の主体として説明する。すなわち、CPU16Bがソフトウェアを実行することによって行う動作については、そのソフトウェアによって実現されるアプリケーションを主体として動作説明を行うことにする。これは、図5に示すJava-AP、JAM(Java Application Manager)、電話帳AP等の各種アプリケーションについても同様である。

#### 【0046】

例えば、ユーザの指示がネイティブAPソフトウェアである通信ソフトウェアの起動を要求するものであれば、OSは通信ソフトウェアを起動して移動機16内にて通信APを実現する。この通信APを用いることで、ユーザは通話相手と通話をすることができます。また、ユーザの指示がネイティブAPソフトウェアである電話帳ソフトウェアの起動を要求するものであれば、OSは電話帳ソフトウェアを起動して移動機16内にて電話帳APを実現する。この電話帳APを用いることで、ユーザは、不揮発性メモリ16Dに記憶された電話帳の内容を示すデータ（以後、電話帳データという）を参照・使用・変更することができます。また、ユーザの指示がネイティブAPソフトウェアであるWebブラウザソフトウェアの起動を要求するものであれば、OSはWebブラウザソフトウェアを起動して移動機16内にてWebブラウザを実現する。

#### 【0047】

また、ユーザの指示がネイティブAPソフトウェアであるJAMソフトウェアの起動を要求するものであれば、OSはJAMソフトウェアを起動して移動機16内にてJAMを実現する。JAMは、移動機16にインストールされているJava-APソフトウェアの一覧をユーザに提示し、ユーザにより指定されたJava-APソフトウェアを起動する。具体的には、JAMに対するユーザの指示がJava-APソフトウェアの起動を要求するものであれば、Java-AP環境ソフトウェアが起動されて移動機16内にJava-AP環境が実現され

、次に、指定されたJava-APソフトウェアが起動されてJava-AP環境内にJava-APが実現される。Java-AP環境は、移動機16のような携帯端末に適した軽量のJava仮想マシンであるKVM (K Virtual Machine) と、Java-APに対して提供されるAPI (Application Interface) とを有する。Java-APに対して提供されるAPIは、トラステッドJava-APソフトウェアによって実現されるJava-AP（以後、トラステッドJava-APという）のみに使用が許可されるトラステッドAPIと、あらゆるJava-APに使用が許可される非トラステッドAPIとに分けられる。

#### 【0048】

JAMは、Java-APの挙動を管理するための機能を実現する。

例えば、JAMは、Java-APソフトウェアが起動される場合には、そのJava-APソフトウェアのSDFについてSDFチェックを行うべきか否かを判断し、これを行うべきと判断した場合にはSDFチェックを行う。このために、JAMは、不揮発性メモリ18Dに記憶される起動回数、経過時間、或いは猶予回数をカウントしたり、クリアするための機能を実現する。また、JAMは、このSDFチェックの結果、SDFが有効であった場合には、Java-APソフトウェアを起動を許可するが、その起動後は、SDF内のパーミッション情報に従ってJava-APの挙動を制限する。また、JAMは、SDFチェックの結果、SDFが失効されていた場合には、Java-APソフトウェアの起動を許可しない。またJAMは、SDFチェックの結果、SDFを更新する必要がある場合には、管理サーバ装置18に再度アクセスしてSDFを更新する。

以上が本実施形態の構成である。

#### 【0049】

##### (2) 動作

次に、上記構成からなるシステムの動作について説明する。

###### (2-1) Java-APソフトウェアのインストール

JAMは、Java-APソフトウェアのダウンロードを要求する指示がWebブラウザから通知されると、Java-APソフトウェアを移動機16にダウンロードし、インストールする処理を行う。この処理の流れを図6に示す。

図6に示されるように、JAMは、まず、ダウンロードしようとするJava-APソフトウェアに対応するADFをCPサーバ装置13からダウンロードする（ステップS11）。具体的には、JAMは、ADF-URLを含むHTTPリクエストを生成・送信し、このHTTPリクエストに対するHTTPレスポンスをCPサーバ装置13から受信してADFを取得する。なお、このときのHTTPリクエストに含まれるADF-URLは、ユーザによって入力されるものであってもよいし、HTML(HyperText Markup Language)によって記述されたホームページに埋めこまれたURLのうちでユーザによって指定されたものであってもよい。JAMは、ダウンロードしたADFの内容に従って、APID、パッケージURL、SDF-URL、ADF-URL等を図4に示すようにして不揮発性メモリ16Dに書き込む。

#### 【0050】

次いで、JAMは、ダウンロードしようとしているJava-APソフトウェアを移動機16にインストール可能か否かを、不揮発性メモリ16Dに記憶されたADFの内容に基づいて判定する（ステップS12）。例えば、JAMはADFに記述されていたjarファイルのサイズと、不揮発性メモリ16D内のjarファイルを記憶可能な空き容量とを比較する等の、従来と同様の基準に従って判定すればよい。

#### 【0051】

ここではインストール可能と判定された場合を想定し（ステップS12；Yes）、次いで、JAMは、ダウンロードしようとするJava-APソフトウェアがトラステッドJava-APソフトウェアであるか否かを判定する（ステップS13）。具体的には、JAMは、ステップS11において記憶した内容にAPID及びSDF-URLが含まれている否かを確認し、記述されていれば、このJava-apソフトウェアに対応するSDFが存在する、即ち、トラステッドJava-APソフトウェアであると判定するし、その記述がなければ非トラステッドJava-APソフトウェアであると判定する。ここではAPID及びSDF-URLが記憶されているので、JAMは、ダウンロードしようとするJava-APソフトウェアがトラステッドJava-APソフトウェアと判定す

る（ステップS13；Yes）。

#### 【0052】

次いで、JAMは、このソフトウェアに対応するSDFの内容を管理サーバ装置18から取得する（ステップS14）。すなわち、JAMは、管理サーバ装置18との間にTCPコネクションを確立し、このコネクションを介して、ADF内に記述されているSDF-URLを含むHTTPリクエストを生成・送信し、このHTTPリクエストに対するHTTPレスポンス（図2参照）を受信して、上記コネクションを切断する。

#### 【0053】

次いで、JAMは、取得したSDFの内容が正当か否かについて判断する（ステップS15）。具体的には、JAMは、取得したSDFの内容が予め定められたフォーマットに従って記述されているか否かとか、SDFに内包されているAPIDと既に不揮発性メモリ16Dに記憶されているAPIDとが一致するか否かといった判定を行い、これらの判定結果が肯定的であれば、取得したSDFの内容は正当であると認め（ステップS15；Yes）、これらの内容を図4に示すようにして不揮発性メモリ16Dに記憶する。

#### 【0054】

次いで、JAMは、CPサーバ装置13からJARファイルをダウンロードする（ステップS16）。具体的には、JAMは、既に記憶しているパッケージURLを含むHTTPリクエストを生成してCPサーバ装置13に送信し、このHTTPリクエストに対するHTTPレスポンスを受信してJARファイルを取得する。

#### 【0055】

そして、JAMは、ダウンロードしたJARファイルを不揮発性メモリ16Dに書き込み、トラステッドJava-APソフトウェアのインストールに係る各種処理を行ってから（ステップS17）、インストールに成功した旨をユーザに通知する（ステップS18）。

以降、JAMは、トラステッドJava-APソフトウェアを実行するに際し、トラステッドJava-APの挙動を監視し、トラステッドAPIの使用を制

限するが、この制限は不揮発性メモリ16Dに記憶されるSDF内のパーミッション情報に従って行われることとなる。

#### 【0056】

なお、上記の処理において、Java-Apソフトウェアがインストール不可能と判断された場合（ステップS12；No）、及び、SDFが正当でないと判断された場合（ステップS15；No）、JAMは、インストールに失敗した旨をユーザに通知するとともに（ステップS20）、移動機16の状態をステップS11以前の状態に戻す。

#### 【0057】

また、上記の処理において、トラステッドJava-Apソフトウェアのインストール処理ではないと判断された場合（ステップS13；No、つまり、非トラステッドJava-SPソフトウェアのインストール処理の場合）、JAMは、通常どおり、ADFに記述されていたパッケージURLに基づいてCPサーバ装置13からJarファイルをダウンロードし（ステップS19）、Java-Apソフトウェアのインストールに係る各種処理を行って（ステップS17）、インストールに成功した旨をユーザに通知する（ステップS18）。

#### 【0058】

##### （2-2）Java-Apソフトウェアの起動

次に、図7に示すフローを参照しながら、Java-Apソフトウェアの起動時の動作について説明する。

JAMは、ユーザによってJava-Apソフトウェアの起動指示がなされると、まず、起動の対象となるJava-ApソフトウェアがトラステッドJava-Apソフトウェアか否かを判定する（ステップS31）。ここでは、トラステッドJava-Apソフトウェアが起動対象であるので（ステップS31；Yes）、処理はステップS32に進み、JAMは、SDFチェックが必要か否かを判断する。具体的には、不揮発性メモリ16Dに記憶されている起動回数がSDFチェック頻度以上か、或いは、経過時間がSDFチェック間隔以上かという条件のうちのいずれか一方が満たされていた場合にはSDFチェックが必要と判断され、上記の条件のいずれもが満たされていない場合にはSDFチェックは不

要と判断される。

#### 【0059】

例えば図4に示された内容では、上記条件のいずれもが満たされていないためSDFチェックは不要と判断され（ステップS32；No）、JAMは、不揮発性メモリ16Dに記憶されている起動回数を1だけインクリメントして（ステップS33）、トラステッドJava-APソフトウェアを起動する（ステップS34）。

#### 【0060】

上記のようにしてJAMがトラステッドJava-APソフトウェアを繰り返し起動していくうちに、例えば起動回数がSDFチェック頻度と同数になると、ステップS32ではSDFチェックが必要と判断される（ステップS32；Yes）。この場合、JAMは、SDFチェックを要求するためのHTTPリクエストを生成し、管理サーバ装置18に送信する。このHTTPリクエストには、起動対象のトラステッドJava-APソフトウェアのAPIIDとSDFの最終更新日時（パラメータ「Lmd」のパラメータ値）とが含まれている。

#### 【0061】

このHTTPリクエストを受信した管理サーバ装置18は、HTTPリクエストからAPIIDを抽出し、このAPIIDを含むSDFを不揮発性メモリ18Aから読み出し、そのSDFステータスを参照する。さらに、管理サーバ装置18は、上記HTTPリクエストから最終更新日時を抽出し、この最終更新日時と不揮発性メモリ18Aから読み出したSDFの最終更新日時とを比較し、移動機16におけるSDFを更新すべきか否かを判断する。つまり、管理サーバ装置18は、HTTPリクエストから抽出した最終更新日時が不揮発性メモリ18Aから読み出したSDFの最終更新日時より古ければ更新要と判断するし、これらが一致すれば更新不要と判断する。上記の処理を経て、管理サーバ装置18は、移動機16に通知すべきSDFステータスの内容を決定し、決定されたSDFステータスを含むHTTPレスポンスを生成し、これをSDFチェック応答として移動機16に送信する。

#### 【0062】

ここで、図8及び図9にSDFチェック応答の内容を例示する。

図8はSDFが有効な状態である場合のSDFチェック応答である。この場合、図8に示すように、SDFステータスを示すパラメータ「Sts」のパラメータ値が有効であることを意味する「00」になっている。一方、図9はSDFが失効されている状態である場合のSDFチェック応答である。この場合、図9に示すように、SDFステータスを示すパラメータ「Sts」のパラメータ値が失効であることを意味する「10」になっている。また、図示はしていないが、SDFが更新されている場合には、パラメータ「Sts」のパラメータ値として、SDFの更新が必要であることを意味する「01」が記述される。また、移動機16によって通知されたAPIDに対応するSDFが管理サーバ装置18によって記憶されていないときは、その旨を示すパラメータ値「99」が記述される。

#### 【0063】

さて、移動機16がSDFチェック応答を受信すると（ステップ36；Yes）、JAMは、そのSDFチェック応答の内容を判断する（ステップS37）。

例えば図8に示すようなSDFチェック応答を受信した場合、JAMは、SDFが有効であると判断し（ステップS37；有効）、不揮発性メモリ16Dに記憶されている起動回数、経過時間及び猶予回数をクリアしてから（ステップS38）、トラステッドJava-APソフトウェアを起動する（ステップS34）。

#### 【0064】

一方、図9に示すようなSDFチェック応答を受信した場合、JAMは、SDFは失効されている状態であると判断し（ステップS37；失効）、SDFが失効状態にあるのでアプリケーションを起動できない旨のメッセージを表示してステップS31以前の状態に戻る処理を行う（ステップS39）。

#### 【0065】

また、SDFステータスが「更新」を意味する場合、JAMは、SDFが更新されている状態であると判断し（ステップS37；更新）、SDFを更新するための処理を行う（ステップS40）。具体的には、前述した図6のステップS14と同様に、管理サーバ装置18からSDFの内容を取得し、さらに、取得した

SDFの内容が正当か否かについて判断した後に、そのSDFの内容を不揮発性メモリ16Dに記憶する。

#### 【0066】

上記の処理において、SDFチェックを行おうとしても圈外状態や通信障害を理由としてそれが不可能であった場合（ステップS36；No）、JAMは、不揮発性メモリ16Dに記憶されている猶予回数がSDFチェック猶予回数未満であるか否かを判断する（ステップS41）。

猶予回数がSDFチェック猶予回数未満である場合には（ステップS41；Yes）、JAMは、不揮発性メモリ16Dに記憶されている猶予回数を1だけインクリメントして（ステップS42）、トラステッドJava-APソフトウェアを起動する（ステップS34）。

これに対し、猶予回数がSDFチェック猶予回数以上である場合には（ステップS41；No）、JAMは、SDFチェックができない旨のメッセージを表示してステップS31以前の状態に戻る処理を行う（ステップS39）。

#### 【0067】

(2-3) トラステッドJava-APソフトウェアを起動するまでのシステム全体の動作例

上記動作の一例を図10を用いて説明する。

図10の説明においては、前述のJAMやJava-AP等のアプリケーションが行う動作は移動機16の動作となることから、動作の主体を移動機16として説明を行う。

図10において、移動機16は、ユーザによってJava-APソフトウェアのダウンロードを要求する指示がなされると、まず、ダウンロードしようとするJava-APソフトウェアに対応するADFのADF-URLを含むHTTPリクエストm1を生成して送信する。

CPサーバ装置13は、このHTTPリクエストに応じて、ADFを含むHTTPレスポンスマ2を生成し移動機16に送信する。

#### 【0068】

このHTTPレスポンスマ2を受信した移動機16は、このADFの内容を不

揮発性メモリ 16D に記憶した後、Java-AP ソフトウェアを移動機 16 にインストール可能と判断すれば、SDF-URL を含む HTTP リクエスト m3 を生成して管理サーバ装置 18 に送信する。

管理サーバ装置 18 は、この HTTP リクエストに応じて、SDF を含む HTTP レスポンス m4 を生成し、移動機 16 に送信する。

#### 【0069】

この HTTP レスポンス m4 を受信した移動機 16 は、この SDF の内容の正当性を確認した後、不揮発性メモリ 16D に記憶する、そして、移動機 16 は、パッケージ URL を含む HTTP リクエスト m5 を生成して CP サーバ装置 13 に送信する。

CP サーバ装置 13 は、この HTTP リクエスト m5 に応じて、jar ファイルを含む HTTP レスポンス m6 を生成し移動機 16 に送信する。

移動機 16 は、受信した jar ファイルを不揮発性メモリ 16D に書き込み、トラステッド Java-AP ソフトウェアをインストールする。

#### 【0070】

この後、移動機 16 は、ユーザによって Java-AP ソフトウェアの起動指示がなされると SDF チェックが必要か否かを判断し、必要と判断すると、APIID と SDF の最終更新日時を含む HTTP リクエスト m7 を SDF チェック要求として管理サーバ装置 18 に送信する。

この HTTP リクエスト m7 を受信した管理サーバ装置 18 は、この HTTP リクエストに含まれる APIID に対応する SDF ステータスを含む HTTP レスポンス m8 を生成し、これを SDF チェック応答として移動機 16 に送信する。

#### 【0071】

この HTTP レスポンスを受信した移動機 16 は、その SDF チェック応答の内容を判断する。例えば SDF チェック応答が SDF の更新を意味する場合、移動機 16 は SDF を更新する。つまり、移動機 16 は、SDF-URL を含む HTTP リクエスト m9 を生成し、管理サーバ装置 18 に送信する。

この HTTP リクエスト m9 を受信した管理サーバ装置 18 は、この HTTP リクエストに含まれる SDF-URL によって特定される SDF の内容を含む H

T T P レスポンス m 1 0 を生成し、これを移動機 1 6 に送信する。

以降、移動機 1 6 は、 J a v a - A P ソフトウェアを起動し、更新された S D F の内容に従って上記 J a v a - A P ソフトウェアを実行する。

#### 【0072】

(2-4) J a v a - A P ソフトウェアが実行されている時の移動機 1 6 の挙動  
次に、上述の各々の J a v a - A P ソフトウェアが実行されている時の移動機 1 6 の挙動について説明する。

##### (2-4-1) 非トラステッド J a v a - A P

上述したインストール動作により移動機 1 6 にインストールされた非トラステッド J a v a - A P ソフトウェアが移動機 1 6 において起動され、このソフトウェアに対応した機能（以後、非トラステッド J a v a - A P）が移動機 1 6 内に実現されたときの移動機 1 6 の挙動について説明する。

#### 【0073】

非トラステッド J a v a - A P が使用しようとする A P I が非トラステッド A P I の場合、前述したように非トラステッド A P I はあらゆる J a v a - A P の使用が許可されているから、この場合の A P I の使用は移動機 1 6 により許可されることとなる。したがって、非トラステッド J a v a - A P はこの非トラステッド A P I を使用することができる。

また、非トラステッド J a v a - A P が使用しようとする A P I がトラステッド A P I の場合、移動機 1 6 はこの J a v a - A P に対応する S D F が不揮発性メモリ 1 6 D に記憶されているか否かを調べる。ここでは、そのような S D F は不揮発性メモリ 1 6 D に記憶されていないから、移動機 1 6 は非トラステッド J a v a - A P によるこの A P I の使用を禁止する。したがって、非トラステッド J a v a - A P はトラステッド A P I を使用することができない。

#### 【0074】

##### (2-4-2) トラステッド J a v a - A P

次に、移動機 1 6 にインストールされたトラステッド J a v a - A P ソフトウェアが、移動機 1 6 が実現された移動機 1 6 において起動され、このソフトウェアに対応した機能が移動機 1 6 内に実現されたときの移動機 1 6 の挙動について

説明する。

トラステッドJava-APが使用しようとするAPIが非トラステッドAPIの場合、前述したように、このAPIの使用は移動機16によって当然許可される。したがって、トラステッドJava-APはこの非トラステッドAPIを使用することができる。

トラステッドJava-APが使用しようとするAPIがトラステッドAPIの場合、このJava-APに対応するSDFが不揮発性メモリ16Dに記憶されているので、このAPIの使用は移動機16によって許可され得るが、そのトラステッドJava-APの挙動はSDF内のパーミッション情報に依存する。

例えば、パーミッション情報のうちのパラメータ「GetPrivateInfo」のパラメータ値が「yes」に設定されている場合、トラステッドJava-APは不揮発性メモリ16Dから電話帳データや未読電子メールを読み出すことができる。また、トラステッドJava-APは、パーミッション情報のうちのパラメータ「AllowedHost」のパラメータ値として設定されているURLの通信装置との間では通信を行うことができる。

#### 【0075】

以上説明したように、移動機16においては、ダウンロードしたSDFに含まれるパーミッション情報の内容に応じた挙動がこのSDFに対応するトラステッドJava-APソフトウェアに許可され、パーミッション情報の内容に含まれていない挙動は許可されない。ユーザから覗れば、従来通りの非トラステッドJava-APの他に、上記のような、より自由な挙動が許可されたトラステッドJava-APを利用可能となり、非常に便利である。

#### 【0076】

なお、上述の通信システムにおいては、移動機16に対し、ADF、SDF、Jarファイルという順序で各種ファイルの配信を行っていたが、このような順序で配信することにより、以下のような効果が生ずる。

既に説明したように、Java-APソフトウェア（ADF及びJarファイル）はCPによって設計・作成され、各々のCPがインターネット上に開設している専用サイト（図1のCPサーバ装置13）において、一般ユーザに公開され

ている。従って、ユーザはまず、CPの専用サイトにアクセスし、そこで、様々なJava-APソフトウェアの解説ページを参照してそのソフトウェアをダウンロードするか否かを判断するのが普通である。そして、ユーザはJava-APソフトウェアをダウンロードしようと判断すると、そのダウンロード処理を指示する操作を行う必要があるが、そのために上記の解説ページには次にダウンロードすべきファイルのURLがアンカータグによって埋め込まれているのが普通である。このとき、CPの立場から視れば、解説ページにADFのURLを埋め込むのが最も手間がかからない。なぜなら、ADFはCPの管理下にあるので、そのADFのURLはCPによって常に把握できているからである。これに対し、解説ページにSDFのURLを埋め込むとなると、CPは通信事業者に問い合わせをする等して、URLの正誤の確認処理を絶えず欠かさないようにしなければならない。よって、ADF、SDF、Jarファイルという順序で各種ファイルの配信を行うことは非常に有意義である。

### 【0077】

また、上記の順序は、現在既に実施されているJava-APソフトウェアのバージョンアップ処理を考慮した場合にも利点がある。現在実施されている処理は、ユーザによってバージョンアップを要求する操作がなされると、移動機は、まず、ADFに記述された内容を参照し、ADFに記述されたパッケージURLに基づいて、バージョンアップ後のJarファイルを取得するという流れで行われている。即ち、バージョンアップ時には、まずADFを参照してから、その後にダウンロード処理に移行するようになっている。

この点を考慮すると、本実施形態の通信システムにおけるバージョンアップ時においても、まずADFを参照し、そのADFに記述されているSDF-URLに基づいてSDFを取得した後、Jarファイルを取得するというように、まずADFの参照から一連の処理を開始すると、それ以降は、SDF→Jarファイルという通常のダウンロードと同じ流れで処理を行うことができ、現状のサービス仕様をあまり変更しないで済む。これに対し、仮にSDF、ADF、Jarファイルという順序で各種ファイルをダウンロードすることが定義付けられている場合、バージョンアップしようとした場合、ADFを参照からダウンロード処理

を開始すると、SDFを取得することなくJarファイルの取得処理にまで至ってしまう。SDFは、バージョンアップ時に書き換えられることは十分にあり得るので、SDFが無いとセキュリティ上で不都合が生ずるおそれがある。以上のような観点からも、ADF、SDF、Jarファイルという順序で各種ファイルの配信を行うことは有意義である。

### 【0078】

#### (3) 変形例

本発明は上述した実施形態に限定されず、以下のような種々の変更が可能である。

上述した通信システムでは、Java-APの挙動を制限する対象としてJava-APが利用するAPIとJava-APの通信相手を挙げたが、本発明はこれに限定されるものではなく、任意の資源（リソース）を対象とすることができる。ここでいう資源はAPIのようなソフトウェア資源であってもよいし、Java-APの通信相手のようなネットワーク資源であってもよいし、ハードウェア資源であってもよい。

ハードウェア資源としては、メモリやスピーカ、マイク、赤外線コントローラ、LED（Light Emitting Diode）等の移動機が備え得るものや、移動機と共に働くUIM（User Identity Module）やSIM（Subscriber Identity Module）等の外部機器なども挙げられる。

### 【0079】

また、ネットワーク資源としては、Java-APの通信相手のようなネットワーク上のリソースの他にも次のようなものも考えられる。前述したように、移動機は移動通信網との間で無線通信を行う。この無線通信時には、移動機は、移動通信網により提供される無線チャネル等の無線資源を使用する。この無線資源はネットワーク資源の一種である。また、移動機は無線資源が属する通信プロトコルレイヤよりも高位の通信プロトコルレイヤにおいて、パケットの伝送路や回線接続の通信路などの通信資源を使用する。このような通信資源もネットワーク資源の一種である。

### 【0080】

次にソフトウェア資源について説明する。ソフトウェア資源としては、前述したA P Iやクラス、パッケージ等が挙げられる。ソフトウェア資源が提供する機能は様々であるが、典型的な機能として、暗号演算などの演算処理機能や、Webブラウザ等の他のアプリケーションとの間でデータを送受したりする機能などが挙げられる。また、本発明は、上記外部機器が有するソフトウェア資源をも利用の制限対象とする態様を技術的範囲に含む。

#### 【0081】

また、上述の実施形態では、JAMは、パーミッション情報に基づいてトラステッドJava-APの挙動を制限するようにしていたが、もし、このトラステッドJava-APがパーミッション情報に相反する挙動をしたときには、このトラステッドJava-APソフトウェアの実行を強制的に終了させるようにしてもよい。

#### 【0082】

また、上述の通信システムではソフトウェアは移動機へ配信されるが、本発明の技術的範囲には、移動機以外の端末装置へ配信する態様も含まれる。

また、上述した通信システムにおいて、信頼できる機関がCPとなってよいこと、すなわち、管理サーバ装置がCPサーバ装置を兼ねるようにしてよいことは言うまでもない。

#### 【0083】

上述した移動機のCPUが実行するソフトウェア（JAMソフトウェアやOSソフトウェア等）は、CPUによって読み取り可能な磁気記録媒体、光記録媒体あるいはROM等の記録媒体に記録して提供することが可能である。また、これらプログラムを、インターネットのようなネットワーク経由で移動機にダウンロードさせることももちろん可能である。

#### 【0084】

##### 【発明の効果】

本発明によれば、移動機のような端末装置においてソフトウェアを起動する場合に、そのソフトウェアによって実現されるアプリケーションに与えられた権限を示すパーミッション情報が有効か否かを外部装置に確認するので、起動すべき

でないソフトウェアを起動することを防止することができる。

【図面の簡単な説明】

【図 1】 本発明の実施の一形態に係る通信システムの構成を示すブロック図である。

【図 2】 同システムにおける SDF の内容を示す図である。

【図 3】 同システムを構成する移動機の構成を示すブロック図である。

【図 4】 同移動機の不揮発性メモリに記憶されている内容の一例を示す図である。

【図 5】 同移動機の機能構成を示す概念図である。

【図 6】 同移動機が Java-AP ソフトウェアをダウンロードしインストールする処理の流れを示すフローチャートである。

【図 7】 同移動機が Java-AP ソフトウェアを起動する処理の流れを示すフローチャートである。

【図 8】 同通信システムにおける SDF チェック応答の一例を示す図である。

【図 9】 同通信システムにおける SDF チェック応答の一例を示す図である。

【図 10】 同通信システムの動作を説明するためのシーケンス図である。

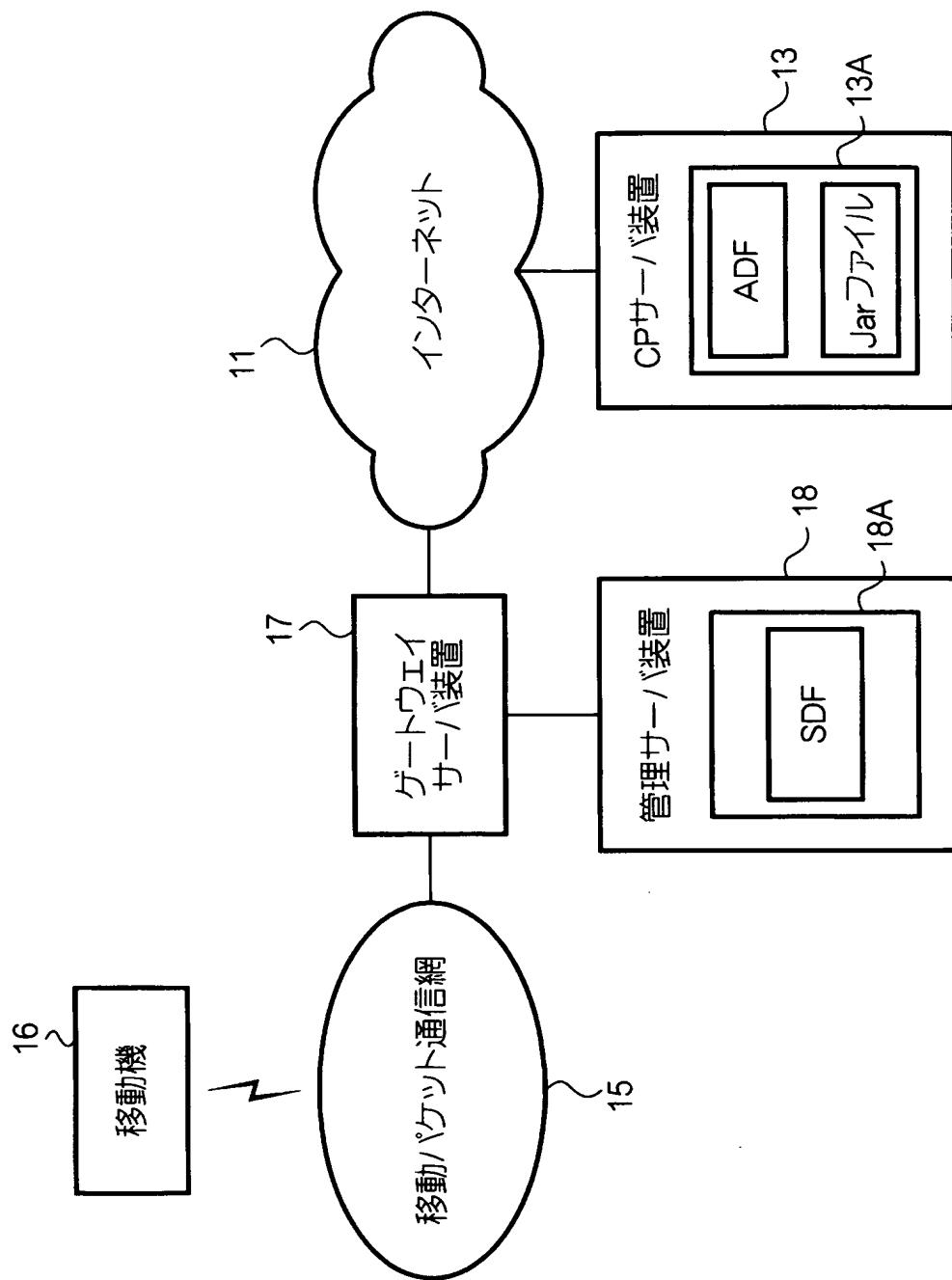
【符号の説明】

11・・・インターネット、13・・・CP サーバ装置（情報提供サーバ装置）、  
15・・・移動パケット通信網、16・・・移動機（端末装置）、17・・・ゲートウェイサーバ装置、18・・・管理サーバ装置（外部装置、管理サーバ装置）、16A・・・ROM、16B・・・CPU（実行手段、チェック手段、起動制御手段、判断手段、起動回数カウント手段、更新手段、終了手段）、16C・・・表示部、16D・・・不揮発性メモリ（記憶手段、起動回数カウント手段、頻度情報記憶手段、間隔情報記憶手段、猶予回数記憶手段）、16E・・・RAM、16F・・・通信部（チェック手段、更新手段）、16G・・・操作部、16H・・・計時部（計時手段）。

【書類名】

図面

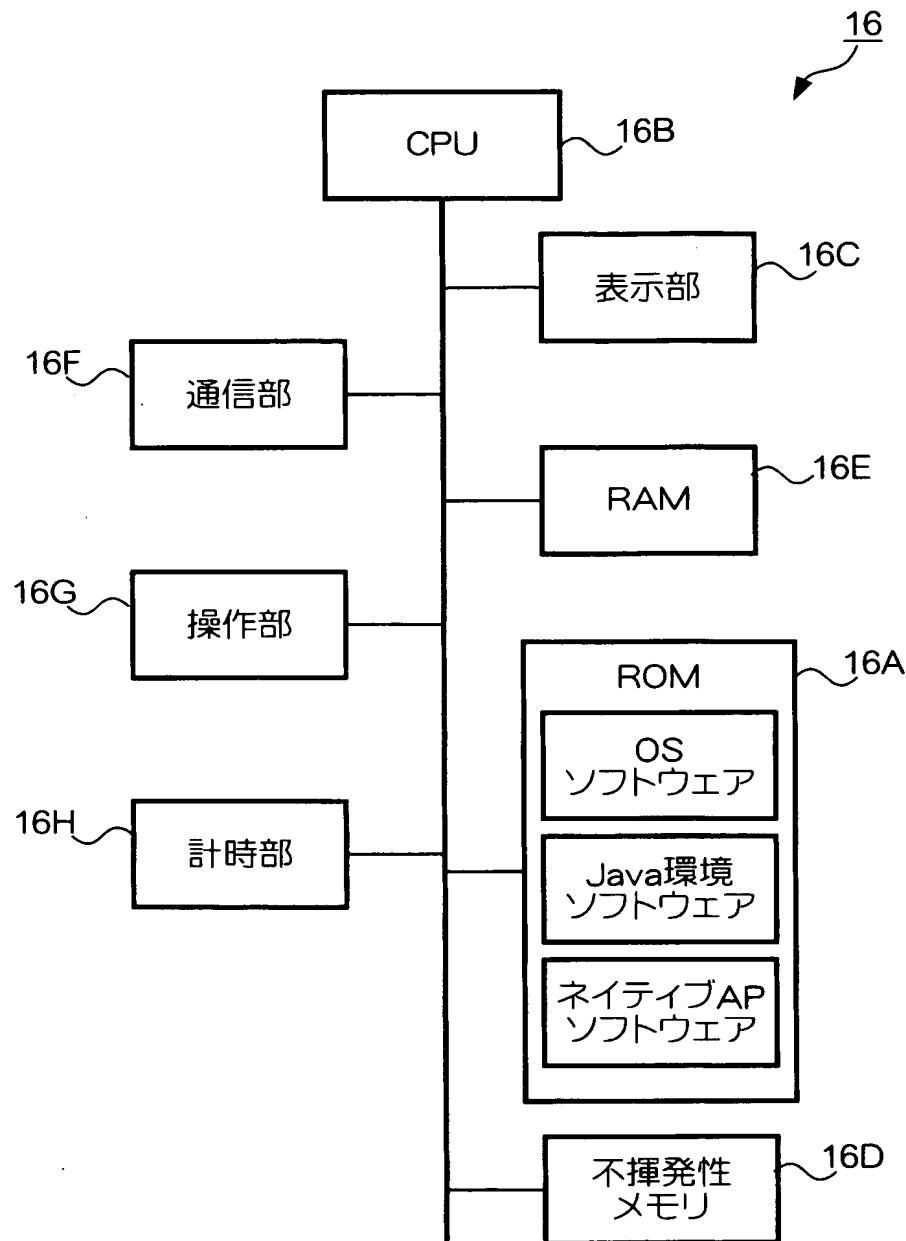
【図 1】



【図 2】

```
HTTP/1.0 200 OK
Content-Type:application/x-sdf
.
.
<CR><LF>
Sts = 00 <CR><LF>
PackageURL = http://cpserver.com:8080/Demo.jar <CR><LF>
CheckCnt = 005 <CR><LF>
CheckInt = 020 <CR><LF>
SuspendedCnt = 005 <CR><LF>
Lmd=20020614120552 <CR><LF>
GetPrivateInfo = yes <CR><LF>
UseMailer = yes <CR><LF>
MessageApp = yes <CR><LF>
SetPhoneTheme = yes<CR><LF>
SetLaunchTime = yes <CR><LF>
AllowedHost = http://aaa.co.jp http://bbb.co.jp:8080 <CR><LF>
```

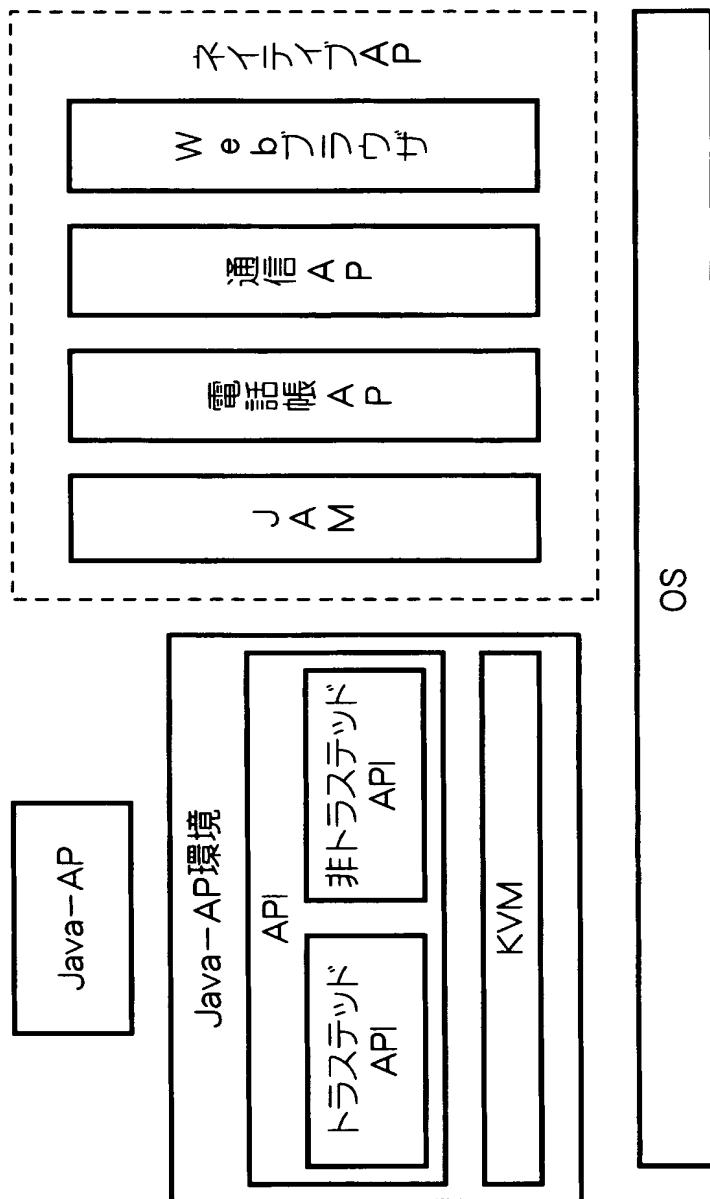
【図3】



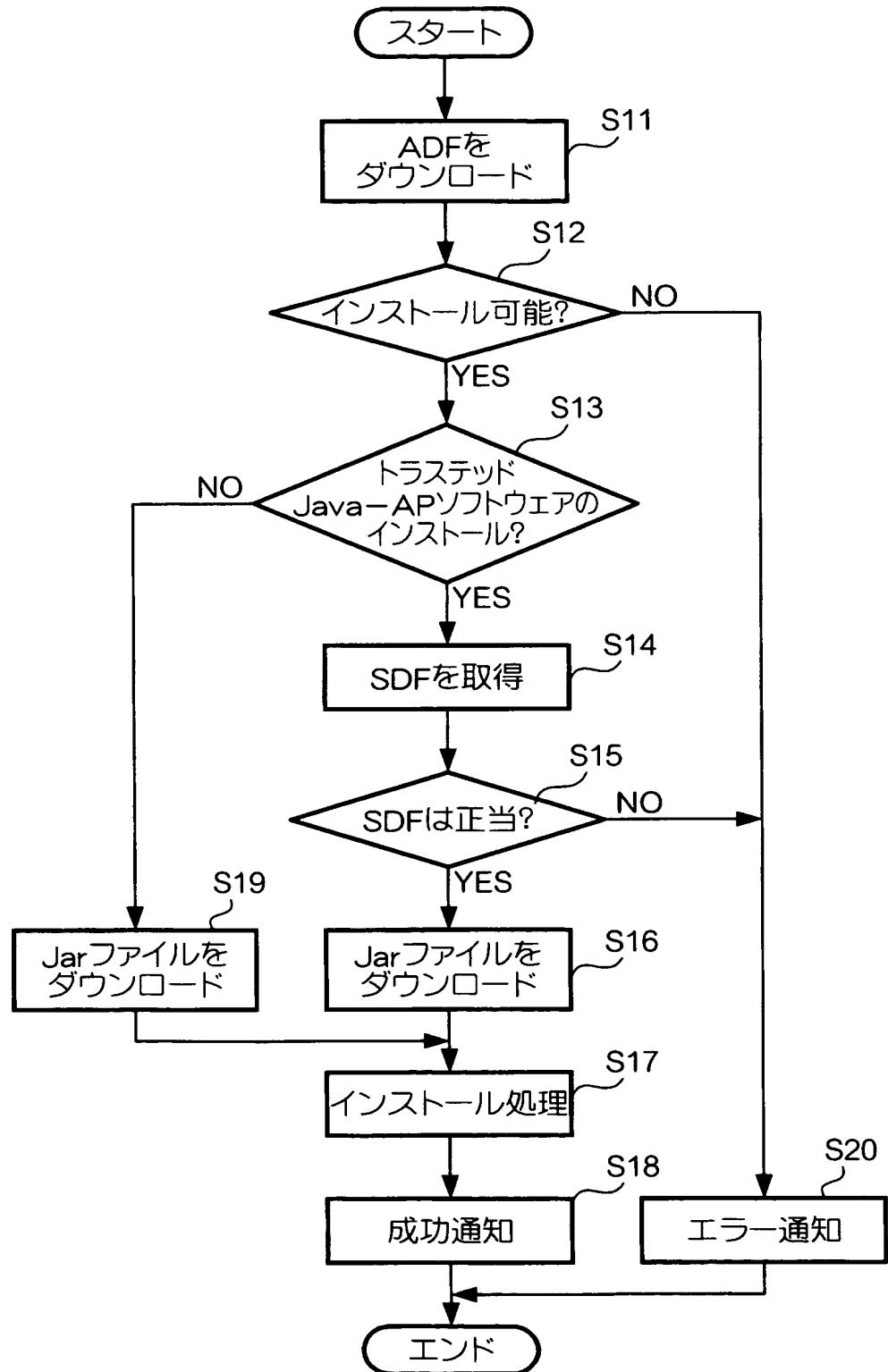
【図 4】

APIID	1000001						
ADF-URL	<a href="http://CPserver.com:8080/Demo.adf">http://CPserver.com:8080/Demo.adf</a>						
SDF-URL	<a href="http://MNserver.com:8080/Demo.sdf">http://MNserver.com:8080/Demo.sdf</a>						
ナビゲーションURL	<a href="http://CPserver.com:8080/Demo.jar">http://CPserver.com:8080/Demo.jar</a>						
SDFステータス	00(有効)						
SDFチェック頻度	5回					起動回数	2回
SDFチェック間隔	20日					経過時間	15日
SDFチェック猶予回数	5回					猶予回数	1回
最終更新日	2002年6月14日12時5分52秒						
ナビゲーション情報	GetPrivateInfo	可					
	Usermailer	可					
	MessageApp	可					
	SetPhoneTheme	可					
	SetLaunchTime	可					
	AllowedHost	<a href="aaa.co.jp:8080">aaa.co.jp:8080</a>					
	AllowedHost	<a href="bbb.co.jp:8080">bbb.co.jp:8080</a>					

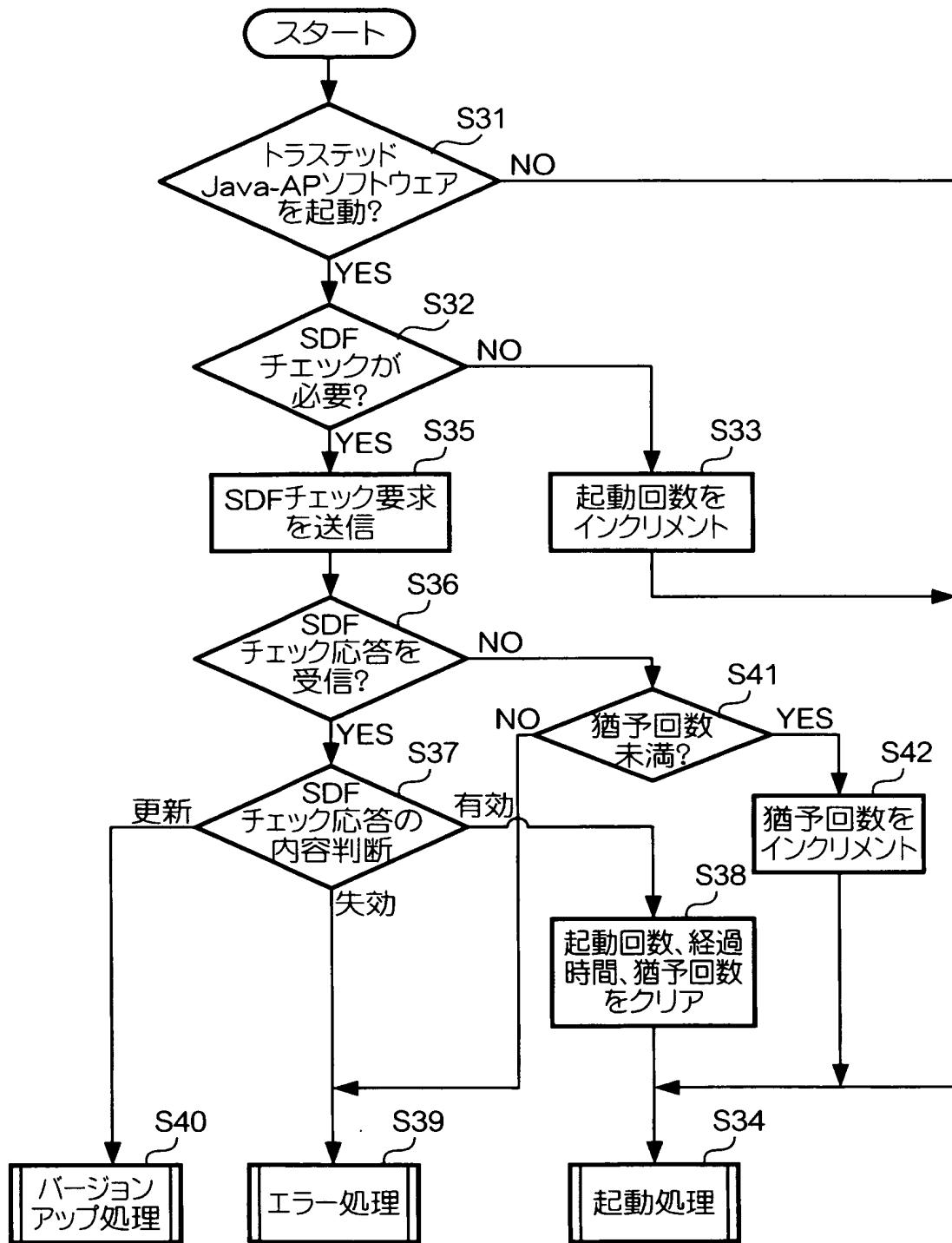
【図5】



【図 6】



【図 7】



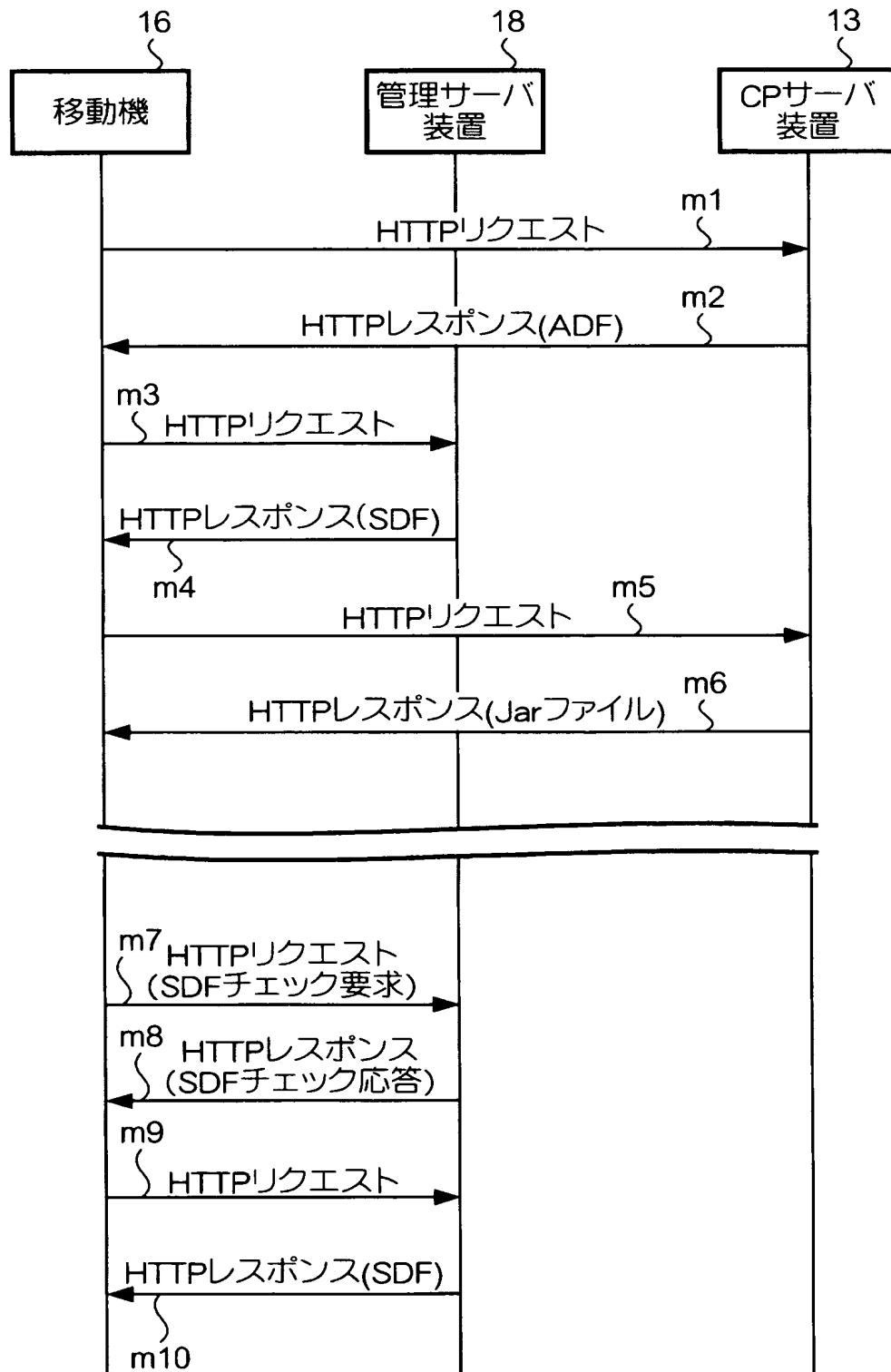
【図8】

```
HTTP/1.0 200 OK
Content-Type:application/x-sdf
.
.
<CR><LF>
Sts = 00 <CR><LF>
```

【図9】

```
HTTP/1.0 200 OK
Content-Type:application/x-sdf
.
.
<CR><LF>
Sts = 10 <CR><LF>
```

【図10】



【書類名】 要約書

【要約】

【課題】 アプリケーションに与えられた権限が変更された場合に、その変更後の権限内容を、移動機等の端末装置におけるアプリケーションに反映させる。

【解決手段】 本システムにおけるJava-APソフトウェアのダウンロードは、ADF、SDF、Jarファイルという順で配信される。SDFとは、移動機内におけるJava-APの挙動を制限する内容が記述されたファイルである。このSDFは有効である状態と失効されている状態とがあり、この有効或いは失効の別は管理サーバ装置18によって記憶されている。移動機は、インストールしたJava-APソフトウェアを起動する際には、まず、上記サーバにアクセスしてSDFの有効或いは失効の状態の別を確認し、SDFが有効な状態であれば、そのSDFの記述内容に従ってJava-APソフトウェアを起動する

【選択図】 図1

特願 2003-096015

出願人履歴情報

識別番号 [392026693]

1. 変更年月日 2000年 5月19日

[変更理由] 名称変更

住所変更

住 所 東京都千代田区永田町二丁目11番1号  
氏 名 株式会社エヌ・ティ・ティ・ドコモ